

AI Tools Usage Documentation

CSE211 – Web Programming

Fall 2025–2026

Final Course Project – Group 01

EventsX

Dynamic Responsive Event Planner Web Application

Team Members

- **Ibrahim Hassan** — 223104567
- **George Karm Hosney Fayz** — 223106365
- **Youssef Ahmed Ibrahim** — 223101109
- **Abdallah Mostafa Mahdy** — 223104683

1. Executive Summary

This document presents a comprehensive overview of the artificial intelligence (AI) tools utilized during the development of the **EventsX** web application. The purpose of this documentation is to ensure transparency and compliance with course guidelines regarding AI usage.

AI tools were used strictly as **learning and development aids**, assisting the team in understanding complex concepts, debugging issues, and exploring unfamiliar technologies. At the beginning of the project, the team had limited experience with backend development and advanced web programming concepts. AI assistance helped bridge these knowledge gaps and accelerated the learning process.

The team emphasizes that **all final implementations, architectural decisions, and code structures were the result of collaborative team effort and full understanding**. Every feature was discussed, implemented, tested, and refined by the team members.

2. AI Tools Used

The following AI tools were used throughout the project development:

2.1 Cursor Built-in AI

Primary Usage:

Cursor's integrated AI assistant was the most frequently used tool, providing real-time support directly within the development environment.

Use Cases:

- Code completion and inline suggestions
- Explanations of complex code blocks
- Debugging syntax and logic errors
- Refactoring suggestions
- Assistance in generating function and module documentation

2.2 Google Gemini

Primary Usage:

Google Gemini was used to understand high-level concepts, architectural ideas, and web development best practices.

Use Cases:

- RESTful API design principles
- MongoDB schema design patterns
- Express.js middleware concepts
- JWT authentication workflows
- Responsive web design techniques

2.3 Antigravity

Primary Usage:

Antigravity was used for detailed technical explanations and troubleshooting more complex issues.

Use Cases:

- Debugging database connection problems
- Understanding and configuring CORS
- Resolving authentication token issues
- Clarifying async/await and promise behavior
- Troubleshooting deployment-related errors

3. Purpose and Usage Patterns

3.1 Understanding New Concepts

A primary purpose of using AI tools was to understand concepts that the team encountered for the first time, particularly in backend development using Node.js, Express.js, and MongoDB.

AI tools helped explain:

- Express.js routing and project structure
- MongoDB schema design and Mongoose ODM
- JWT token structure and authentication flow
- Middleware implementation
- Async/await and promise handling
- CORS configuration for frontend-backend communication

Team Contribution:

After receiving explanations, team members discussed the concepts, experimented with implementations, and adapted the knowledge to fit the project's specific requirements. All final decisions were made collaboratively.

3.2 Applying New Technologies

When implementing unfamiliar technologies, AI tools provided initial guidance on:

- Express.js server setup
- Password hashing using bcryptjs
- MongoDB model creation and validation
- Cookie-based JWT authentication
- RESTful API endpoint structure
- Error-handling middleware

Team Contribution:

The team modified all suggested approaches, tested multiple solutions, and integrated them with existing code. Every feature was reviewed and refined collectively.

3.3 Debugging and Troubleshooting

AI tools were particularly useful during debugging phases, helping identify potential causes of errors such as:

- JavaScript syntax and logic errors
- MongoDB connection and query issues
- CORS-related frontend/backend errors
- JWT validation and authentication issues
- Async/await promise rejections
- Cookie parsing and token storage problems

Team Contribution:

Team members first analyzed errors independently, then used AI tools for clarification. All fixes were implemented, tested, and validated by the team.

3.4 File Creation and Documentation Assistance

AI tools assisted in creating initial documentation structures, including:

- README.md templates

- Requirements and documentation outlines
- Suggested documentation sections
- Code comment and JSDoc templates

Team Contribution:

All documentation content was written, reviewed, and customized by the team. AI tools only provided structural guidance; all project-specific content was original.

3.5 Backend Development Support

Given the team's limited backend experience, AI tools were especially helpful in understanding:

- Express.js server configuration
- Controller, route, and model separation
- Database connection patterns
- Request validation and sanitization
- Role-based access control
- API status codes and response formatting

Team Contribution:

All backend logic, database schemas, and architectural decisions were designed and implemented by the team through discussion and testing.

4. Specific AI-Assisted Components

4.1 Backend Server Setup

- **AI Assistance:** Guidance on Express.js server structure and environment configuration
- **Team Work:** Custom middleware, CORS configuration, routing integration, and error handling were fully implemented by the team

4.2 Database Models

- **AI Assistance:** Understanding Mongoose schema syntax and validation
- **Team Work:** All schemas, relationships, and validation logic were designed and implemented by the team

4.3 Authentication System

- **AI Assistance:** JWT structure, bcrypt usage, and authentication patterns
- **Team Work:** Full authentication flow, password rules, token handling, and route protection were implemented by the team

4.4 API Endpoints

- **AI Assistance:** RESTful routing conventions
- **Team Work:** Endpoint design, business logic, validation, and error handling were entirely team-driven

4.5 Frontend–Backend Integration

- **AI Assistance:** Understanding API calls and token handling
- **Team Work:** All frontend logic, UI design, authentication flow, and error handling were implemented by the team

5. Learning Outcomes

Through responsible use of AI tools, the team gained hands-on experience in:

- Backend development with Node.js and Express.js
- MongoDB and Mongoose database design
- JWT-based authentication and authorization
- RESTful API development

- Frontend–backend integration
- Debugging and error handling
- Secure and organized code structuring

Team members can now independently explain, modify, and extend all project components.

6. Code Ownership and Originality

All code in this project is **original work** by the team. The team:

- Designed the complete application architecture
- Made all implementation decisions
- Wrote, tested, and debugged all code
- Customized all solutions to meet project requirements
- Conducted thorough testing and quality assurance

AI tools were used similarly to documentation or online references—not as code generators.

7. Ethical Considerations

The team confirms that AI tools were used in full compliance with course guidelines:

- AI tools were used for learning and guidance only
- All code was written and understood by the team
- All design and architectural decisions were team-made
- The project reflects genuine learning and skill development

8. Conclusion

AI tools played a supportive role in accelerating learning and overcoming initial technical challenges, particularly in backend development. However, the successful completion of

EventsX is the result of **collaborative effort, deep understanding, and hands-on implementation** by all team members.

The final product reflects the team's creativity, technical growth, and commitment to ethical AI usage. All team members actively contributed and can confidently explain and modify every part of the application.